

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	11	712/9.ccls. and (vector near2 register) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/15 09:24
L2	7	712/9.ccls. and (vector near2 register) and simd and (pointer or "indirect addressing")	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/15 09:25
S1	26	((vector\$6 or array or pipelin\$3 or simd or matrices) near5 loop\$3) and (access\$4 near5 pattern) and (array or pipelin\$3 or simd or matrices or vector\$6) near5 register) and (reorder\$3 or rearrang\$4 or restructur\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 09:18
S2	2	"20050097301"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 11:10
S3	74	access\$3 same (constant near3 stride)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 09:42
S4	31	"vector pointer register"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 15:51
S5	1	S1 and S4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 09:18
S6	25	access\$3 same (constant near3 stride) and ((vector\$6 or array or pipelin\$3 or simd or matrices) near5 loop\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 10:41
S7	9	config\$7 same "vector pointer register"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 10:07

EAST Search History

S8	236	(BEN-DAVID or NAISHLOS or SHVADRON or ZAKS) and ((memory near3 access\$3) or loop\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 10:08
S9	49	(BEN-DAVID or NAISHLOS or SHVADRON or ZAKS).in. and ((memory near3 access\$3) or loop\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 10:09
S10	0	("2004/0015677").URPN.	USPAT	OR	OFF	2007/01/23 10:17
S11	4	("20040006681" "5669010" "5850227" "6665790").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2007/01/23 10:18
S12	5	("6665790").URPN.	USPAT	OR	OFF	2007/01/23 10:23
S13	4317	simd or simdd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 11:10
S14	569	(simd or simdd) and loop\$3 same register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 11:10
S15	2	(simd or simdd) same loop\$3 same register same (reorder\$3 or rearrang\$5 or reconfig\$7)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 13:05
S16	6	("6665790" "6915411").pn. "20040078554"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 15:40
S17	4	"6446105".pn. "20050055536"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 15:40
S18	181	717/160.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 15:52

EAST Search History

S19	137	717/161.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:23
S20	506	717/151.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:23
S21	1247	711/5.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:23
S22	329	712/34.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:23
S23	259	712/35.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:14
S24	316	712/22.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:14
S25	646	712/228.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:22
S26	500	712/207.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:14
S27	3	712/207.ccls. and (simd or simdd) and loop\$3 same register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:14
S28	4	712/228.ccls. and (simd or simdd) and loop\$3 same register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:14

EAST Search History

S29	48	712/22.ccls. and (simd or simdd) and loop\$3 same register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:23
S30	10	712/35.ccls. and (simd or simdd) and loop\$3 same register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:23
S31	6	711/5.ccls. and (simd or simdd) and loop\$3 same register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:23
S32	7	717/151.ccls. and (simd or simdd) and loop\$3 same register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:23
S33	1	717/161.ccls. and (simd or simdd) and loop\$3 same register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/02/06 16:23
S34	1	(simd or "single instruction multiple data") and (vector near3 register) and multiport and "scalar register file"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/07 08:40
S35	9	(configur\$3 near2 register) and multiport and scalar and (vector\$3 or array)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 16:06
S36	1	(multiport and scalar) near3 register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 16:07
S37	26	(configur\$5 or reconfigur\$5) near3 register and (multiport and scalar)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 16:08
S38	7	(configur\$5 or reconfigur\$5) near3 register and (multiport and scalar) and (vector near2 element)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:01

EAST Search History

S39	3	(configur\$5 or reconfigur\$5) near3 register and (multiport and scalar) and (configur\$5 or reconfigur\$5) near3 vector\$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:06
S40	0	("reorder on read") or ("reorder on write") same (vector near3 file) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:07
S41	0	((("reorder on read") or ("reorder on write")) same (vector near3 file) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:07
S42	0	((("reorder on read") or ("reorder on write")) and (vector near3 file) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:07
S43	0	((("reorder on read") or ("reorder on write")))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:12
S44	8	(indirect\$2 adj address\$3) near3 (vector near2 register)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:16
S45	2	717/140.ccls. and (vector near2 register) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:17
S46	2	717/149.ccls. and (vector near2 register) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:18
S47	1	717/150.ccls. and (vector near2 register) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:19
S48	2	717/160.ccls. and (vector near2 register) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:32

EAST Search History

S49	5	712/10.ccls. and (vector near2 register) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:32
S50	3	712/220.ccls. and (vector near2 register) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/15 09:22
S51	51	712/22.ccls. and (vector near2 register) and simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/14 17:32


[Web](#) [Images](#) [Video](#) [News](#) [Maps](#) [more »](#)

[Advanced Scholar Search](#)
[Scholar Preferences](#)
[Scholar Help](#)

Scholar [All articles](#) - [Recent articles](#) Results 1 - 10 of about 118 for **vector pointer register multiport**

All Results

[F Catthoor](#)
[H De Man](#)
[L Nachtergaele](#)
[F Franssen](#)
[J Tseng](#)

Banked multiported **register** files for high-frequency superscalar microprocessors - all 16 versions »

JH Tseng, K Asanovic - Computer Architecture, 2003. Proceedings. 30th Annual ..., 2003 - [ieeexplore.ieee.org](#)

... A banked **multiport register file** can provide sufficient bandwidth for ... the 4-issue pipeline with **register file** of size ... de- pendent on the stack **pointer** tend to ...

Cited by 75 - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

A high-performance embedded DSP core with novel SIMD features - all 5 versions »

JH Derby, JH Moreno - Acoustics, Speech, and Signal Processing, 2003. Proceedings. ..., 2003 - [ieeexplore.ieee.org](#)

... **vector pointer** mechanism, which permits the addressing in a very flexible way of groups of four 16-bit ele- ments in a large, **multiport, scalar register file**. ...

Cited by 6 - [Related Articles](#) - [Web Search](#)

Vectorizing for a SIMdD DSP architecture - all 5 versions »

D Naishlos, M Biberstein, S Ben-David, A Zaks - Proceedings of the international conference on Compilers, ..., 2003 - [portal.acm.org](#)

... **File** (VEF) — a large, **multiport, scalar register file** containing 2 N ... from the **Vector Element File** (VEF) is ... indices specified by **Vector Pointer** Registers (VPRs ...

Cited by 14 - [Related Articles](#) - [Web Search](#)

An innovative low power high-performance programmable signal processor for digital communications. - all 5 versions »

JH Moreno, V Zyuban, U Shvadron, FD Neeser, JH ... - IBM Journal of Research and Development, 2003 - [research.ibm.com](#)

... four elements selected from the large **multi-port register file**. ... has been preloaded into **scalar register** sr0, which ... for auto-updating the target **vector pointer**. ...

Cited by 19 - [Related Articles](#) - [Cached](#) - [Web Search](#) - [BL Direct](#)

A LOW-POWER HIGH-PERFORMANCE EMBEDDED DSP CORE WITH NOVEL SIMD FEATURES

JH Derby, JH Moreno, MS Ware - [research.ibm.com](#)

... four data elements from a large **multiport** array of ... At the same time, the **Vector Pointer**

Registers are ... ments are placed on another **register file**, wherein each ...

[Related Articles](#) - [View as HTML](#) - [Web Search](#)

Vectorization in a SIMdD DSP architecture

S Ben-David, D Naishlos, U Shvadron, A Zaks - 2005 - [freepatentsonline.com](#)

... **Element File** (VEF) 112, typically being a **multiport, scalar register file** containing independently ... Thus, for example, a **vector pointer** setup by the tuple [0044 ...

[Cached](#) - [Web Search](#)

Method and apparatus for **vector** processing - all 3 versions »


VA Desai, DP Gurney, B Chau - US Patent 6,922,716, 2005 - Google Patents

... example contains a 16-bit **scalar** arith- 45 ... L/SB—one associated with each **vector** 60 processing ... invention, includes a 32x16-bit address **pointer register file**. ...
[Related Articles](#) - [Web Search](#)

Extending a Prolog architecture for high performance numeric computations
R Yung, AM Despain, YN Patt - System Sciences, 1989. Vol. I: Architecture Track, ..., 1989
- [ieeexplore.ieee.org](#)
... the entire numeric structure addressed by an indirect **pointer** will not be transferred into the PLM **register** set, but ... CC 01 2 2 2 2 FS 1 **vector** length (arity ...
[Web Search](#)

An integrated prolog architecture for symbolic and numeric executions
R Yung, AM Despain - Annals of Mathematics and Artificial Intelligence, 1991 - Springer
... the entire numeric structure addressed by an indirect **pointer** will not be ... Floadi transfers an element from a **vector register** into a **scalar register**. ...
[Related Articles](#) - [Web Search](#)

An innovative low-power high-performance programmable signal processor - all 4 versions »
A Zaks, A Geva, S Ben-David, SW Asaad, TW Fox, D ... - [research.ibm.com](#)
... Instruction storage Branch unit Integer unit Storage access unit **Vector pointer** unit ... 32 64 **Scalar** units **Vector** units Condition **register** logic (8 bits) ...
[Related Articles](#) - [View as HTML](#) - [Web Search](#)

Google 

Result Page: 1 2 3 4 5 6 7 8 9 10 [Next](#)

vector pointer register multiport scalar register file

[Google Home](#) - [About Google](#) - [About Google Scholar](#)

©2007 Google



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Published since January 1995 and Published before October 2003

Found 4 of 75,649

Terms used: vector pointer register multiport scalar reorder

Sort results by

☒ Save results to a Binder
Try an [Advanced Search](#)Try this search in [The ACM Guide](#)

Display results

☒ Search Tips

☐ Open results in a new window

Results 1 - 4 of 4

Relevance scale ☐ ☐ ☐ ☐ ☐ ☒1 [A survey of processors with explicit multithreading](#)

Theo Ungerer, Borut Robič, Jurij Silc

March 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 1

Publisher: ACM Press

Full text available: pdf(920.16 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Hardware multithreading is becoming a generally applied technique in the next generation of microprocessors. Several multithreaded processors are announced by industry or already into production in the areas of high-performance microprocessors, media, and network processors. A multithreaded processor is able to pursue two or more threads of control in parallel within the processor pipeline. The contexts of two or more threads of control are often stored in separate on-chip register sets. Unused i ...

Keywords: Blocked multithreading, interleaved multithreading, simultaneous multithreading

2 [Data and memory optimization techniques for embedded systems](#)

P. R. Panda, F. Catthoor, N. D. Dutt, K. Danckaert, E. Brockmeyer, C. Kulkarni, A.

Vanderappelle, P. G. Kjeldsberg

April 2001 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**,

Volume 6 Issue 2

Publisher: ACM Press

Full text available: pdf(339.91 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present a survey of the state-of-the-art techniques used in performing data and memory-related optimizations in embedded systems. The optimizations are targeted directly or indirectly at the memory subsystem, and impact one or more out of three important cost metrics: area, performance, and power dissipation of the resulting implementation. We first examine architecture-independent optimizations in the form of code transformations. We next cover a broad spectrum of optimizati ...

Keywords: DRAM, SRAM, address generation, allocation, architecture exploration, code transformation, data cache, data optimization, high-level synthesis, memory architecture customization, memory power dissipation, register file, size estimation, survey

3 Instruction path coprocessors



Yuan Chou, John Paul Shen

May 2000 **ACM SIGARCH Computer Architecture News**, **Proceedings of the 27th annual international symposium on Computer architecture ISCA '00**, Volume 28 Issue 2

Publisher: ACM Press

Full text available: [pdf\(134.64 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents the concept of an Instruction Path Coprocessor (I-COP), which is a programmable on-chip coprocessor, with its own mini-instruction set, that operates on the core processor's instructions to transform them into an internal format that can be more efficiently executed. It is located off the critical path of the core processor to ensure that it does not negatively impact the core processor's cycle time or pipeline depth. An I-COP is highly versatile and can be used ...

4 Improving single-process performance with multithreaded processors



Alexandre Farcy, Olivier Temam

January 1996 **Proceedings of the 10th international conference on Supercomputing ICS '96**

Publisher: ACM Press

Full text available: [pdf\(1.07 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

Results 1 - 4 of 4

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)